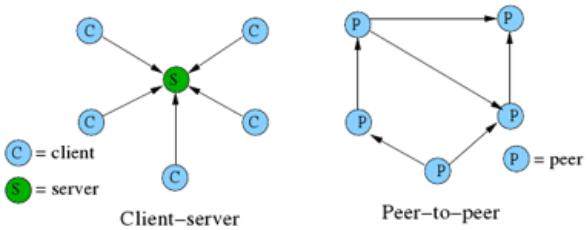
La Question Technique : ça marche comment Internet ? (partie 2/3)

Deuxième partie (sur trois) de La Question Technique consacrée au réseau des réseaux : Internet. Après nous être intéressés aux bases techniques la semaine dernière, cap aujourd'hui sur les protocoles applicatifs. Parce qu'une fois qu'on a mis en place l'infrastructure réseau initiale, il faut bien s'en servir...

Si vous avez - par malheur - manqué le précédent volet de La Question Technique, un petit rattrapage pourrait s'avérer nécessaire en cliquant ici, pour découvrir la première partie de cet article consacré au fonctionnement d'Internet. Sinon, la suite, c'est par ici...

Connecter des millions de machines entre elles et leur permettre d'échanger des paquets de données est une chose, leur permettre de se comprendre en est une autre… Les protocoles TCP et IP sont des protocoles très génériques, qui définissent la structure des paquets et des éventuelles données de contrôle, mais qui ne donnent pas de règles sur le contenu des paquets et leur interprétation. Cette tâche incombe à des protocoles de plus haut niveau : les protocoles « applicatifs ».

Ceux-ci se répartissent principalement en deux catégories : les protocoles client-serveur et les protocoles P2P.



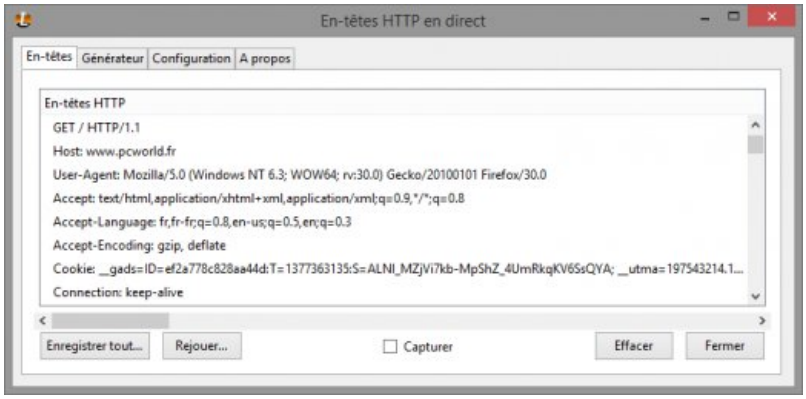
Les protocoles client-serveur

Les protocoles client-serveur sont les plus anciens et les plus nombreux. Ce sont ces protocoles qui sont utilisés pour fournir la plupart des services sur Internet : sites web, messagerie, streaming (il existe toutefois des protocoles de streaming P2P), cloud…
Dans une architecture client-serveur, il y a pour chaque service un petit nombre de serveur (dans la suite, on considérera qu'il n'y en a qu'un) et un grand nombre de clients. Les clients ne communiquent jamais directement entre eux, toutes les requêtes qu'ils envoient sont envoyées vers le serveur. Par exemple, si Alice veut envoyer un e-mail à Bob, il ne sera pas transmis directement de la machine d'Alice à celle de Bob, il passera par au moins un serveur de messagerie.

C'est l'architecture la plus simple d'un point de vue réseau, et elle a l'avantage de très bien s'accommoder du routage NAT du côté du client, ce dernier étant alors totalement transparent. Le NAT est aussi possible côté serveur, à condition de configurer correctement le routeur.

HTTP

Le HTTP (HyperText Transfer Protocol) est un protocole très simple, qui ne propose qu'une poignée de commandes, dont seulement deux sont utilisées dans l'écrasante majorité des cas. Il a été conçu au milieu des années 90, principalement pour la consultation de sites web, ce qui en fait aujourd'hui l'un des protocoles les plus utilisés.



Exemple de requête HTTP vers pcworld.fr

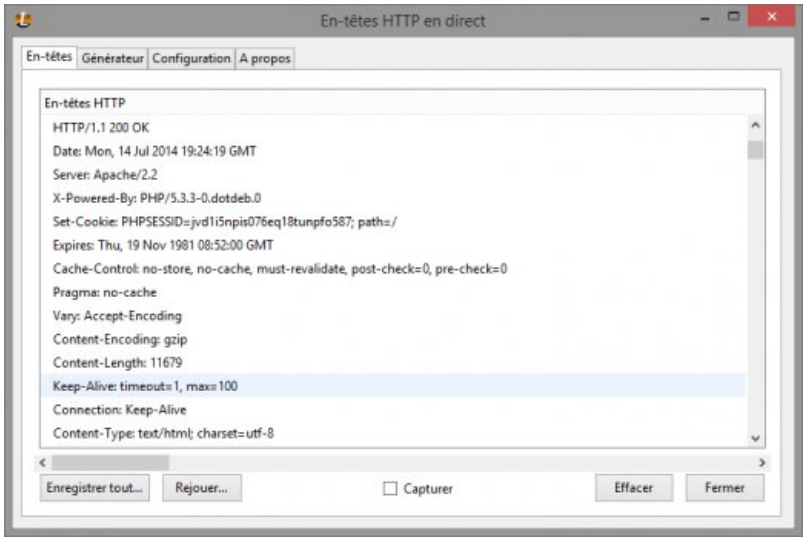
Une requête HTTP et un paquet commençant par si simple texte. La première ligne contient la commande avec ses éventuels paramètres obligatoires (par exemple, le fichier demandé avec une requête GET) et la version du protocole (1.1 actuellement), les lignes suivantes contiennent des paramètres optionnels que le client décide d'envoyer au serveur. Par exemple, quand le client accède au serveur via un nom DNS plutôt que son IP, il renseigne dans la requête un paramètre « Host » mentionnant ce nom DNS (ainsi, plusieurs noms DNS peuvent être hébergés sur un même serveur, la requête du client précisant quel est le site concerné). Il communique généralement aussi différentes informations sur ses capacités techniques ainsi que les éventuels cookies (une liste de clés associées à une valeur) définis lors d'une visite précédente sur le site. Ces paramètres complémentaires peuvent également contenir des informations d'identification (nom d'utilisateur et mot de passe) pour l'accès à certains fichiers lorsque le contrôle d'accès est fait au niveau du serveur HTTP.

Les deux commandes les plus courantes sont la commande GET, qui sert à récupérer le fichier passé en paramètre et POST qui sert à soumettre des données au serveur pour traitement (par exemple, le contenu d'un formulaire).

Il existe aussi des commandes permettant d'envoyer des fichiers à un endroit précis sur le serveur ou de supprimer un fichier du serveur, mais elles sont plus rarement utilisées, les modifications sur le contenu d'un site étant généralement plutôt effectuées avec d'autres protocoles plus spécialisés (FTP par exemple).

L'en-tête de la requête se termine par une ligne vide, indiquant au serveur que la suite de la requête correspond aux données éventuellement transmises avec la requête.

Les serveurs HTTP attendent généralement les requêtes sur le port 80 ou le port 443 si la connexion est chiffrée.



La réponse à la requête précédente

Comme la requête, la réponse commence par un en-tête texte. Sa première ligne contient le code re réponse (2xx en cas de succès, 3xx pour les redirections, 4xx pour les erreurs, comme la très connue erreur 404 indiquant que le fichier demandé n'a pas été trouvée), souvent accompagné d'un petit texte descriptif (par exemple « Not found » pour une erreur 404). Les lignes suivantes de l'en-tête donnent différentes informations sur le contenu de la réponse : date de dernière modification, date d'expiration

en cas de mise en cache, informations sur la possibilité de mise en cache (notez que même si le serveur indique que le contenu ne doit pas être mis en cache, le client peut très bien passer outre…), taille, type, encodage…

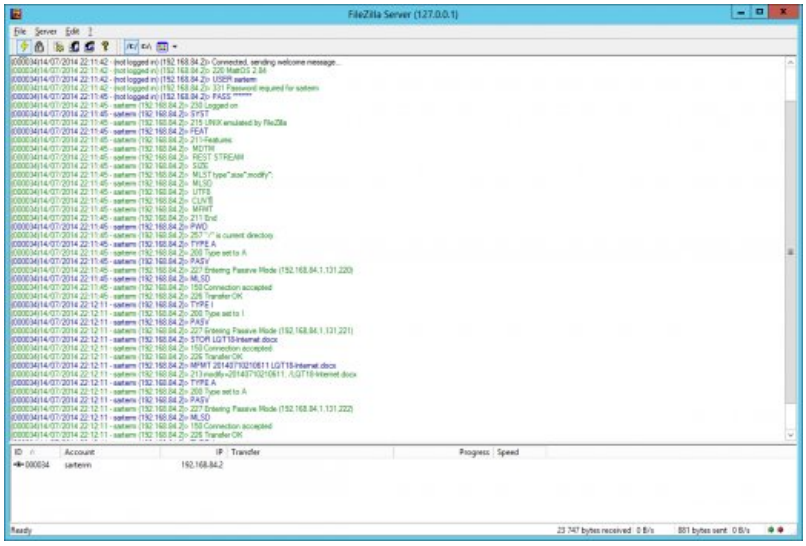
L'en-tête se termine par une ligne vide, indiquant au client que la suite correspond aux données.

La simplicité du protocole HTTP et le fait qu'il passe sans difficultés à travers les NAT et la plupart des firewall en configuration par défaut font que sa plage d'utilisation s'est largement étendue au fil du temps, et il est aujourd'hui loin de se limiter à la consultation de pages web. Il est par exemple beaucoup utilisé pour de la communication entre applications (webservices), en remplaçant les pages web par des données structurées compréhensibles par l'application cliente (XML, JSON…).

FTP

Autre protocole « pilier » d'Internet (sa première version, datant de 1971, est même largement antérieure à IP, publié en 1981), le FTP (File Transfer Protocol) a été conçu comme son nom l'indique pour l'échange de fichiers entre un client et un serveur. Il propose donc un panel de commandes plus large que celui du http, pour être plus efficace dans cet usage, par exemple la commande MGET qui permet de récupérer plusieurs fichiers en une seule requête (Multiple GET).

Il s'agit par ailleurs d'un protocole à session : l'utilisateur commence par se connecter au serveur avec une première requête, généralement sur le port 21, puis s'y identifie si nécessaire et peut ensuite envoyer plusieurs requêtes sans recommencer à chaque fois la connexion et l'identification (en HTTP, chaque requête est indépendante de la précédente et les informations d'identification sont envoyées à chaque fois, les sessions sont gérées à un niveau plus élevé, via les cookies).



Exemple de session FTP, en bleu les requêtes du client, en vert les réponses du serveur

Dans son mode de fonctionnement par défaut, le protocole FTP a également la particularité d'inverser le rôle du client et du serveur lors du transfert des données. En effet, alors que la connexion utilisée pour les commandes se fait dans le sens classique, du client vers le serveur, le transfert des données se fait dans une seconde connexion, du serveur vers le client, après que ce dernier ait indiqué au serveur l'IP et le port sur lequel il est en écoute.

Ce mode de fonctionnement est problématique quand le client est derrière un NAT, puisqu'il faut alors que la passerelle soit configurée pour rediriger certains ports vers le client et que le client communique ces ports ainsi que l'IP publique de la passerelle au serveur. Il rend également FTP inutilisable sur un poste client équipé d'un firewall bloquant toutes les connexions entrantes.

FTP propose heureusement aussi un mode dit « passif », dans lequel la connexion de données s'effectue du client vers le serveur, sur une IP et un port alloué dynamiquement à chaque connexion.

Du fait que les échanges de données en FTP se font sur des ports et IP spécifiés dans les commandes, et non pas sur ceux correspondant à la connexion de commandes, il est également possible d'utiliser FTP pour des transferts directs entre deux serveurs, le client n'étant là que pour passer des commandes. Par exemple, s'il envoie une commande RETR (réception d'un fichier) en mode passif à un serveur puis une commande STOR (envoi d'un fichier) en mode actif au second serveur, en lui communiquant l'IP et le port donnés par le premier serveur suite à la commande GET, le

second serveur ira se connecter au premier pour chercher les données.

POP, SMTP et IMAP

Avec HTTP et FTP, POP et SMTP ont longtemps constitués les quatre protocoles de base quasiment indispensable pour profiter convenablement de sa connexion Internet, même s'ils ont aujourd'hui perdu de leur importance côté client en étant supplantés par les webmails.

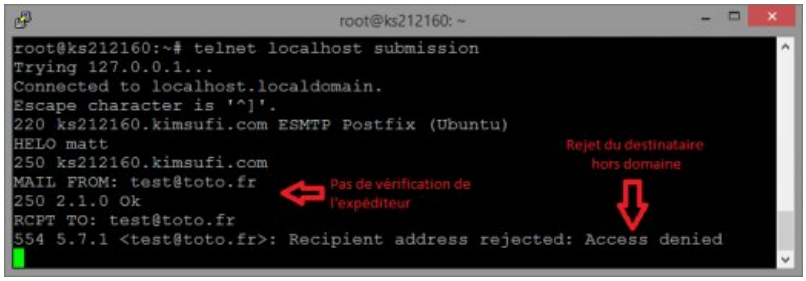
SMTP (Simple Mail Transfer Protocol) est le protocole de base pour l'envoi de courriers électroniques. Il est utilisé à la fois pour le dépôt des messages, l'expéditeur se connectant à son serveur SMTP pour y déposer le message et pour leur transport, les serveurs SMTP échangeant entre eux les messages lorsque c'est nécessaire. Par exemple, lorsque vous vous connectez au serveur SMTP de votre FAI pour envoyer un message vers une adresse @gmail.com, le serveur de votre FAI va se connecter au serveur SMTP de Gmail pour lui faire suivre le message. Chaque serveur intermédiaire ajoutera une entrée dans l'en-tête du message, qui permettra au destinataire de retracer le chemin qu'a pris le message.

```
Delivered-To: destinataire@destinataire.fr
Received: by 10.58.13.131 with SMTP id h3csp41999vec;
        Fri, 11 Jul 2014 08:14:40 -0700 (PDT)
X-Received: by 10.180.211.101 with SMTP id nb5mr5709893wic.53.1405091678762;
        Fri, 11 Jul 2014 08:14:38 -0700 (PDT)
Return-Path: <expediteur@expediteur.fr>
Received: from mail.comparhaut.net (mail.comparhaut.net. [87.98.189.172])
        by mx.google.com with ESMTPS id wd7si4613838wjc.60.2014.07.11.08.14.22
        for <destinataire@destinataire.fr>
        (version=TLSv1 cipher=RC4-SHA bits=128/128);
        Fri, 11 Jul 2014 08:14:22 -0700 (PDT)
Received: from unknown (HELO ALIENPC) (expediteur@expediteur.fr@*.*.*.*)
  by 0 with ESMTPA; 11 Jul 2014 15:14:06 -0000
From: "Mathieu CHARTIER" <expediteur@expediteur.fr>
To: "'Matthieu Sarter'" <destinataire@destinataire.fr>
In-Reply-To: <CAKLGONEFXgbbHGmXoceNFz67oFT=9VQbQwmTUo2zrEQXdecZzg@mail.gmail.com>
Subject: RE: LQT
Date: Fri, 11 Jul 2014 17:14:03 +0200
Message-ID: <029f01cf9d1a$c0807bb0$41817310$@pcworld.fr>
MIME-Version: 1.0
Content-Type: multipart/alternative;
    boundary="----=_NextPart_000_02A0_01CF9D2B.840C31E0"
X-Mailer: Microsoft Outlook 14.0
Thread-Index: AQMPHzCK/eibnNxxptWDg/r5+jRmcAKmtsbNAgD8sYECW27MKgLQZcHBAsCxsAICHFfK4
Content-Language: fr

This is a multipart message in MIME format.

------=_NextPart_000_02A0_01CF9D2B.840C31E0
Content-Type: text/plain;
    charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

Hello !
```

Un message, avec son long en-tête

C'est un protocole très basique, qui dans sa version standard ne permet même pas l'authentification de l'expéditeur. Il suffit en effet de se connecter à un serveur SMTP et de lui spécifier une adresse d'expéditeur (commande MAIL FROM) une adresse de destination (commande RCPT TO) et le contenu du message avec son en-tête (commande DATA, suivi du message en mode texte, y compris pour les pièces jointes, qui doivent être encodées en base 64) pour que le message soit envoyé au destinataire.

Cette absence d'authentification rend les mails peu fiable (il est difficile d'être certain que l'expéditeur est bien celui dont l'adresse est indiquée) et favorise le spam, n'importe qui pouvant se connecter à un serveur SMTP pour lui demander d'envoyer des messages à de nombreux destinataires. Pour remédier à ces limitations, il existe une variante de SMTP imposant une authentification, mais elle est rarement utilisée pour des questions de compatibilité, tandis que l'utilisation de DKIM (DomainKeys Identified Mail) permet d'ajouter dans les messages un champ d'en-tête contenant une signature cryptographique assurant que l'expéditeur du message s'est bien authentifié auprès du serveur d'expédition. Par ailleurs, les serveurs sans authentification sont de plus en plus souvent configurés pour n'accepter que les messages à destination d'un domaine dont ils ont la gestion, ce qui empêche de les utiliser comme relais pour du spam.



Exemple de session SMTP infructueuse avec un serveur n'assurant pas le relais

POP (Post Office Protocol) est pour sa part le protocole historique pour récupérer des courriers électroniques sur un serveur de messagerie. Comme SMTP, il fonctionne intégralement en mode texte, mais cette fois avec une authentification obligatoire, puisque le serveur a besoin de savoir quelle boîte mail l'utilisateur souhaite relever. Son jeu de commandes comporte le strict minimum pour la gestion des mails : affichage du nombre de messages, de leur liste, récupération partielle ou complète d'un message donné (avec son en-tête dans les deux cas, c'est ensuite au client mail de faire le tri et de déterminer ce qu'il doit afficher ou non à l'utilisateur) et suppression d'un message.

Du fait de son fonctionnement très minimaliste, POP est surtout utilisé pour relever les mails quand on ne le fait que depuis une machine, en les supprimant ensuite du serveur. On lui préfère aujourd'hui IMAP (Internet Message Access Protocol), bien plus adapté aux usages modernes (utilisation de plusieurs appareils).

La philosophie d'IMAP est très différente, puisque le serveur ne sert plus seulement à la relève des messages, mais aussi à leur stockage et leur tri, en offrant la possibilité de créer des dossiers sur le serveur pour y ranger les messages. Il permet également d'attacher des métadonnées aux messages, en particulier leur statut de lecture. Ainsi, lorsqu'un utilisateur connecte plusieurs de ses appareils à une même boîte aux lettres IMAP, un message lu sur un appareil apparaitra automatiquement comme lu sur les autres appareils, alors qu'en POP il faudrait le marquer comme lu sur chaque appareil.

Les protocoles pair-à-pair (P2P)

Dans un protocole P2P, la notion de client et de serveur s'efface quasiment, au profit de la notion de nœud : chaque appareil peut fonctionner simultanément en mode client, pour envoyer des requêtes à d'autres nœuds, ou en mode serveur pour recevoir des requêtes d'autres nœuds.

L'intérêt principal de l'architecture P2P est de permettre de transférer des grandes quantités de données sans passer par un serveur unique qui serait vite saturé.
Les réseaux P2P sont généralement plus robustes face aux attaques (la perte de quelques nœuds n'empêche pas le service de fonctionner), à condition de pouvoir se passer d'un serveur central (qui fait office d'annuaire des nœuds connectés), permettent une meilleure répartition du trafic, en cherchant les données sur le nœud le plus proche possible et permettent des latences plus faibles dans les communications, en réduisant le nombre d'intermédiaires.

Ils offrent par contre moins de garanties sur la disponibilité des ressources, qui peut varier en fonction des nœuds connectés au réseau à un instant donné, et sur leur « qualité » (par exemple, sur un réseau d'échange de fichiers, il n'y a pas d'autorité centrale qui valide les fichiers disponibles). Ils peuvent également poser des problèmes aux nœuds situés derrière une passerelle NAT, puisque celle-ci devra être configurée correctement pour permettre le fonctionnement du nœud en mode serveur.

Les protocoles P2P sont surtout connus du grand public pour l'échange de fichiers (BitTorrent, eDonkey), au point que pour beaucoup P2P et échange de fichiers sont synonymes, mais on trouve aussi beaucoup d'architecture P2P dans le domaine des télécommunications temps-réel en audio ou vidéo (par exemple, le protocole de téléphonie SIP peut être utilisé en P2P, tout comme Jingle, l'extension Jabber permettant de gérer des communications audio et vidéo).

Le réseau Tor utilise pour sa part une architecture P2P pour anonymiser des communications client-serveur classiques (les requêtes transitent par différents nœuds du réseau Tor avant d'être envoyées au serveur, qui verra donc l'IP du dernier nœud au lieu de celle de l'émetteur de la requête).

Enfin, on utilise parfois un réseau P2P comme « accélérateur » pour une infrastructure client-serveur. Par exemple, un site web à fort trafic contenant beaucoup de contenu statiques (images, CSS…) peut intégrer dans ses pages un système qui va permettre à un client de charger ces contenus statiques à partir du cache d'autres clients ayant visité le site plutôt qu'à partir du serveur.

Le protocole Telnet

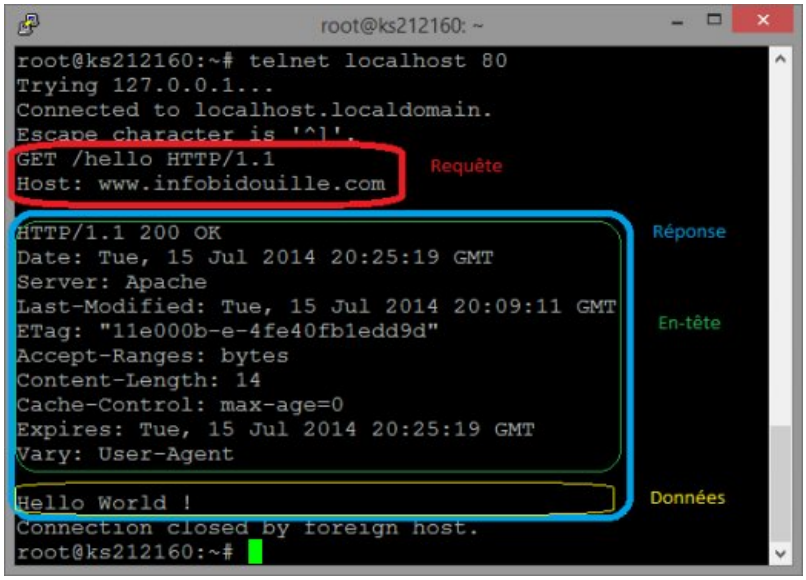
Avant de clore cette seconde partie du dossier sur le fonctionnement d'Internet, il reste un dernier protocole à évoquer, car il va vous permettre par un usage détourné d'approfondir la découverte des protocoles décrits plus haut. Il s'agit du protocole Telnet.

À la base, le protocole Telnet est un protocole simple pour ouvrir une console distante sur un serveur et y exécuter des commandes. Il est généralement remplacé aujourd'hui par le protocole SSH, qui offre une plus grande sécurité, la connexion étant chiffrée.

Mais le protocole Telnet est tellement simple que le client se contente en fait d'envoyer au serveur tout ce que l'utilisateur tape au clavier et d'afficher à l'écran tout ce que le serveur renvoie, ce qui permet de détourner un client Telnet pour l'utiliser comme client pour n'importe quel protocole en mode texte (HTTP, POP, SMTP, etc…), en connectant simplement le client au serveur sur le bon port, plutôt que d'utiliser le port Telnet par défaut.

C'est un excellent moyen pour découvrir le fonctionnement d'un protocole texte en écrivant des requêtes et en observant comment le serveur répond.

Voici par exemple comment on peut utiliser un client Telnet pour interroger un serveur HTTP :



Vous en savez désormais un peu plus sur les protocoles applicatifs. Les exemples donnés ci-dessus ne sont toutefois que la partie émergée de l'iceberg, il existe des milliers de protocoles applicatifs différents, et il serait impossible de tous les lister ici, mais les protocoles de base traités ci-dessus couvrent aujourd'hui la majorité des usages grand public.

La troisième partie de ce dossier traitera de la gestion du trafic entre les différents opérateurs, sur le plan technique, mais aussi sur le plan commercial, qui fait tant parler de lui quand les opérateurs se tirent dans les pattes…

Sur le même thème

+ La Question Technique : ça marche comment Internet ? (partie 1/3)
+ La Question Technique : ARM contre x86, quelles différences fondamentales ?

COMMENTAIRES

Aucun commentaire sur cet article pour le moment, soyez le premier !

VOTRE COMMENTAIRE

suivre ce fil de discussion

Votre commentaire...

Identifiez-vous ou **inscrivez-vous** pour commenter.

Publié le 19 juillet 2014 - Par Matthieu Sarter

Aucun commentaire

PARTAGER

Partager Tweeter

Partager Partager

Connexion Inscription

POUR NOUS SUIVRE...

PCWorld.fr sur Facebook PCWorld.fr sur Google+ PCWorld.fr sur Twitter

newsletter votre@email.com ok

ACTUALITÉS

PCWorld Macworld

samedi 19 juillet

10:00 • LQT : ça marche comment Internet ?

vendredi 18 juillet

17:03 • Intel est en forme, AMD reste optimiste
16:32 • KickStarter : le MagSafe pour iPhone
16:23 • Un pack Quakecon sur Steam
15:31 • Google publie d'excellents résultats
15:04 • iPhone 6 : batterie et capteur photo
14:37 • Un écran Ultra HD de 28" à 499€
13:41 • Android Wear et les cadrans alternatifs
13:34 • Susan Wagner au CA d'Apple
13:08 • Facebook teste un bouton "Buy"
12:05 • De nouveaux Freeplugs à 500 Mbps
12:02 • iTunes Pass disponible en France
11:44 • MS abandonne les Nokia sous Android
11:08 • Une app Facebook pour les célébrités
10:32 • La PS4 devant la XOne malgré tout
10:29 • Touchgrind BMX gratuit sur Mac
09:58 • DOTA 2 en direct à la TV américaine
09:35 • La 4G arrive dans le forfait RED 3 Go
09:07 • L'Apple A9 gravé par Samsung en 2015
08:34 • Megadrive : une (fausse) réédition

jeudi 17 juillet

17:11 • Du concret pour le smartphone modulaire
17:04 • **Test : JXD S5800**
16:37 • Combien de YouTubers payés ?
16:32 • Les clones d'iPhone 6 déjà en vente
16:05 • LG annonce le G3 Beat
15:27 • Microsoft annonce 18.000 licenciements
15:16 • GTX 800MX, Maxwell à tous les étages ?
15:08 • Logitech : de nouveaux accessoires
14:29 • FreedomPop : le MVNO 100% gratuit
13:47 • APU Carrizo et Stacked DRAM
13:26 • iBooks Store : affaire à 450 millions $
13:17 • Microsoft fait la guerre aux Chromebooks
12:12 • Un ex-dictateur attaque Activision
11:53 • Thunderbolt : lecteur cartes P2 Sonnet

11:20 • Une GTX 750 Ti STRIX chez Asus
10:46 • Le MT6595 affole AnTuTu Benchmark
10:33 • Des albums de McCartney en app iPad
09:52 • Une "route électrique" en Chine...
09:08 • Le stylet Pencil de FiftyThree en France
09:05 • La Xbox One se vend 2x mieux sans

mercredi 16 juillet

17:12 • Les nano-pixels pour réinventer la HD ?
16:41 • Le jeu PC plus gros que les consoles
16:32 • CarPlay : Apple convoite Volkswagen
16:07 • La Tesla Model III coûtera $35.000
15:21 • Project Zero : les hackers de Google
15:10 • Flappy Bird débarque sur Apple II
14:14 • La fiche technique du LG G3 mini
13:29 • Google+ et le droit aux pseudos
13:16 • Trois modèles pour l'iWatch
12:26 • Un spatioport anglais pour le tourisme

Nous suivre :

Notre réseau :
News Mac, iPad & iPhone - Prix informatique - Comparateur de prix - Prix Electroménager
Nos partenaires :
PCWorld Roumanie

ComparHaut SARL © 2000-2014 | Tous droits réservés à propos | conditions d'utilisation | contact